

COMP 1010- Summer 2015 (A01)

Jim (James) Young

young@cs.umanitoba.ca

jimyoung.ca

block moving randomly around a screen

A block that jumps around

Globals

- block position

- block size

- how fast it can move

Block moving randomly...

What is random? How much the block moves.
Not the block position

```
float moveX = random(MAX_MOVE);
```

Then we add the movement to the position

```
blockX = blockX + moveX;
```

The block can only move right!

How can we also make it move negative?

```
random(blockSpeed*2)-blockSpeed;
```

if blockspeed is 5..

```
random(5*2)-5;
```

- Generate double the range .
- Subtract the range from it.
- If random gives us 0
 - 0-5 is -5
- If random gives us 5
 - 5-5 is 0
- If random gives us 9
 - 9-5 is 4

It moves off the edge of the screen...

Use min and max to cap it!

Make it ALSO move toward the mouse!!

X first...

How to get how much toward the mouse?

$\text{mouseX} - \text{blockX}$

if mouse is to the right, we get how many

if mouse is to the left, we get neg. how many

How to move toward the mouse?

add the difference to the block

Oh no! We just move TO the mouse

Move **toward** the mouse

Divide the difference by some number....

common basic arithmetic

Processing has a bunch of arithmetic shortcuts that makes your life easier... as long as you understand them

common basic arithmetic

it is very common in programming to increase, decrease, or multiply a variable by a certain amount:

```
variable = variable + 5;
```

```
variable = variable * 10;
```

```
variable = variable / 4;
```

combined operations

variable = variable + 5; variable = variable % 2;

 variable += 5; variable %= 2;

variable = variable * 10;

 variable *= 10;

variable = variable / 4;

 variable /= 4;

variable = variable - 5;

 variable -= 5;

examples..

```
price = price * SALES_TAX;
```

```
price *= SALES_TAX;
```

```
posX = posX - moveX;
```

```
posX -= moveX;
```

```
payment = payment / NUMBER_OF_PEOPLE;
```

```
payment /= NUMBER_OF_PEOPLE;
```

another shorthand: incrementors and decrementors

straight forward: what does increment and decrement mean?

`variable = variable + 1; // increment variable`

`variable = variable - 1; // decrement variable`

one short hand: `variable += 1; variable -= 1;`

incrementor: `variable++`

decrementor: `variable--`

incrementor / decrementor example

```
int i = 5;
```

```
i++;
```

```
i++;
```

```
i--; // what does i equal?
```

```
int i = 5;
```

```
i++; // i now equals 6
```

```
i++; // i now equals 7
```

```
i--; //i now equals 6
```

trivia: why is c++ called c++?

a *very* popular programming language was called C, but it did not have object oriented programming.

some people updated C to include object oriented programming, and called it c++. it means c + 1 😊



Boolean!!



(not bullion)



the **boolean** data type

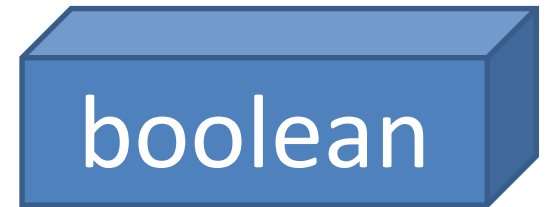
note: the boolean data type stores only two different values: **true** and **false**.

```
boolean isStudentSleeping;
```

```
isStudentSleeping = true;
```

```
boolean jimIsRich = false; // :(
```

true, false



true, false are not numbers.. they're booleans.

but what use is this?? just storing true and false??

we need to learn two things to make booleans useful:

conditionals: code that only runs if a boolean is true or false, e.g.,

```
if (jimsIsRich) { stealHisMoney(); }
```

boolean operations : calculations that use and result in booleans!

the boolean data type is used in computer logic – it gets confusing fast



Logic: another thing that penguins aren't very good at.

conditionals!!

using booleans to have code run
ONLY if certain conditions are met

so far, our programs run in a straight path:

start at the top
work down.

```
int ballSize = 5;
```

```
ballSize += 10;
```

```
ballSize -= 10;
```

```
ellipse(posX,posY,ballSize,ballSize)
```

what if we only want
to grow or shrink the
ball, but not both?
Depending on
circumstances?

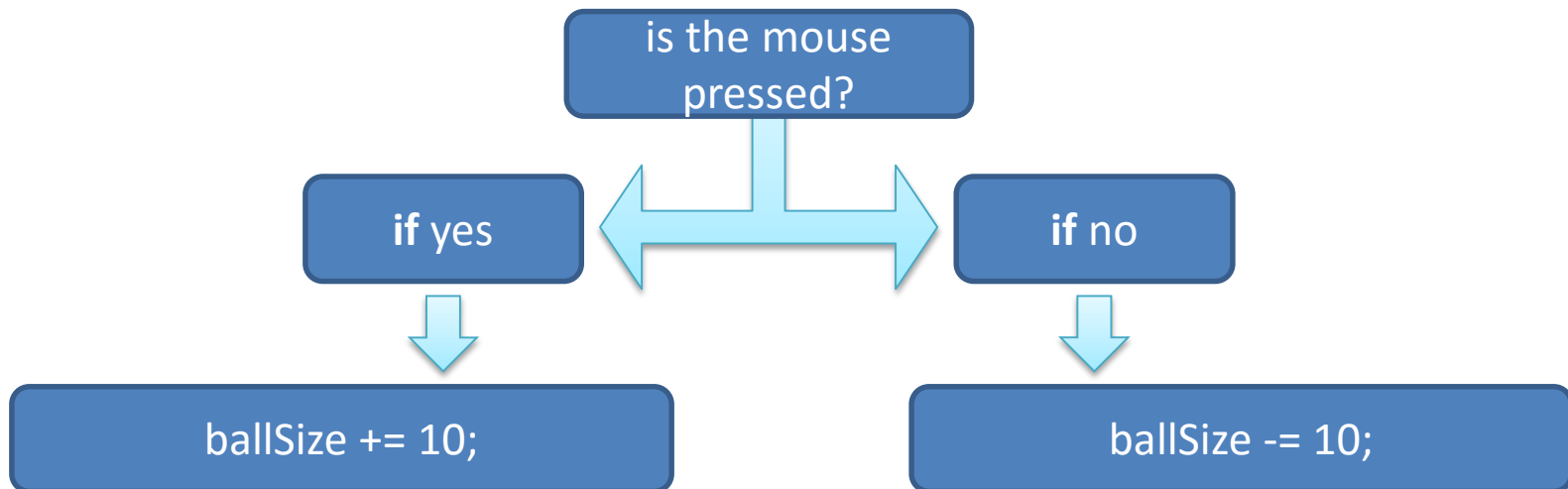
what if we only want
to draw the ball if it is
not too small?

a fork in the road!

we need a way to choose which lines of code are run and which are not run,

and under what conditions they are run

this is **conditional programming**, and achieved using **if** statements



the **if** statement!

Processing uses the **if** keyword, and a **boolean** test, and executes a block of code IF AND ONLY IF the boolean test is true.

```
// .. your program...
```

```
if (booleanValue)
```

```
{
```

```
    // then do this code
```

```
}
```

```
// more program...
```



this code is ONLY run if the booleanValue is true. otherwise, it is skipped completely!!

Processing has some great pre-defined
booleans

```
boolean mousePressed;
```

```
boolean keyPressed;
```

e.g.,

```
if (mousePressed)
```

```
{
```

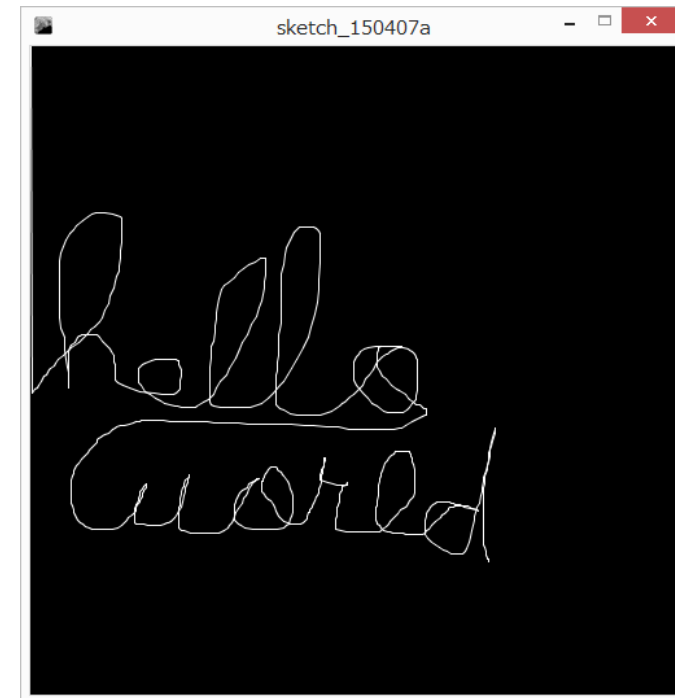
```
    // do stuff
```

```
}
```

Update our drawing program

Only draw when the mouse is pressed

Erase the screen when the keyboard is pressed



Draw a dashed line!!

- Have two drawing colors – A and B.
 - One is black one is white
 - Each time we draw, we swap A and B

```
int colorA =255;
```

```
int colorB = 0;
```

How do we swap the values in two variables???

Swap version1

```
int colorA =255;
```

```
int colorB = 0;
```

```
// swap
```

```
colorA = colorB;    colorA set to 0
```

```
colorB = colorA;    color b set to 0
```

What happens?

To swap a variable, you need a temporary holding variable

```
int colorA =255;
```

```
int colorB = 0;
```

```
int swapTmp;
```

```
// swap
```

```
swapTmp = colorA;    save colorA copy (255)
```

```
colorA = colorB;    colorA is now 0
```

```
colorB = swapTmp;   colorB set to colorA copy(255)
```